| | Type | L # | Hits | Search Text | DBs | Time Stamp | Comments |
|---|---|---|---|---|---|---|---|
| 1 | BRS | L1 | 17 | software WITH hardware WITH prefetch$5 near3 instruction | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:39 | |
| 2 | BRS | L2 | 3 | data adj stream adj touch and VMX | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:15 | |
| 3 | BRS | L3 | 5 | block adj size WITH stride WITH field WITH bit | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:16 | |
| 4 | BRS | L4 | 4 | 3 not 1 | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:25 | |
| 5 | BRS | L5 | 159 | hardware WITH prefetch$5 near3 instruction | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:39 | |
| 6 | BRS | L6 | 30 | 5 and data adj stream | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:39 | |
| 7 | BRS | L7 | 30 | 6 and cache | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:40 | |

| | Type | L # | Hits | Search Text | DBs | Time Stamp | Comments |
|---|---|---|---|---|---|---|---|
| 8 | BRS | L8 | 24 | 7 not 1 | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:40 | |
| 9 | BRS | L9 | 21 | 8 not 4 | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:40 | |
| 10 | BRS | L10 | 1 | 9 and DST | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:40 | |
| 11 | BRS | L11 | 0 | 9 and VMX | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:41 | |
| 12 | BRS | L12 | 5 | 9 and (field WITH bit WITH value) | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:42 | |
| 13 | BRS | L13 | 0 | 9 and (field WITH bit WITH value WITH predetermined) | USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB | 2004/08/28 10:42 | |

Terms used
**prefetch** and **instruction** and **cache** and **mapping** and **hardware** and **block size** and **stride** and **vector** and **DST** and **VMX**

Found **6,889** of **141,680**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Stride directed prefetching in scalar processors

John W. C. Fu, Janak H. Patel, Bob L. Janssens

December 1992 **ACM SIGMICRO Newsletter , Proceedings of the 25th annual international symposium on Microarchitecture**, Volume 23 Issue 1-2

Full text available: pdf(1.36 MB) Additional Information: full citation, references, citings, index terms

2 Data prefetch mechanisms

Steven P. Vanderwiel, David J. Lilja

June 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 2

Full text available: pdf(172.07 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The expanding gap between microprocessor and DRAM performance has necessitated the use of increasingly aggressive techniques designed to reduce or hide the latency of main memory access. Although large cache hierarchies have proven to be effective in reducing this latency for the most frequently used data, it is still not uncommon for many programs to spend more than half their run times stalled on memory requests. Data prefetching has been proposed as a technique for hiding the access lat ...

**Keywords:** memory latency, prefetching

3 A data cache with multiple caching strategies tuned to different types of locality

Antonio González, Carlos Aliagas, Mateo Valero

July 1995 **Proceedings of the 9th international conference on Supercomputing**

Full text available: pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

4 DSTRIDE: data-cache miss-address-based stride prefetching scheme for multimedia processors

Hariprakash. G, Achutharaman. R, Amos R. Omondi

January 2001 **Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture**, Volume 23 Issue 4

Full text available: pdf(928.14 KB) Publisher Site Additional Information: full citation, abstract, references

h c g e cf c

Prefetching reduces cache miss latency by moving data up in memory hierarchy before they are actually needed. Recent hardware-based stride prefetching techniques mostly rely on the processor pipeline information (e.g. program counter and branch prediction table) for prediction. Continuing developments in processor microarchitecture drastically change core pipeline design and require that existing hardware-based stride prefetching techniques be adapted to the evolving new processor architectures. ...

5 Quantifying loop nest locality using SPEC'95 and the perfect benchmarks

Kathryn S. McKinley, Olivier Temam

November 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 4

Full text available: pdf(635.63 KB) Additional Information: full citation, abstract, references, citings, index terms

This article analyzes and quantifies the locality characteristics of numerical loop nests in order to suggest future directions for architecture and software cache optimizations. Since most programs spend the majority of their time in nests, the vast majority of cache optimization techniques target loop nests. In contrast, the locality characteristics that drive these optimizations are usually collected across the entire application rather than at the nest level. Researchers have studied nu ...

6 Speculative prefetching

Y. Jégou, O. Temam

August 1993 **Proceedings of the 7th international conference on Supercomputing**

Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings, index terms

A hardware prefetching mechanism named Speculative Prefetching is proposed. This scheme detects vector accesses issued by a load/store instruction and prefetches the corresponding data. The scheme requires no software add-on, and in some cases it is more powerful than software techniques for identifying regular accesses. The tradeoffs related to its hardware implementation are extensively discussed in order to finely tune the mechanism. Experiments show that average memory ...

7 A quantitative analysis of loop nest locality

Kathryn S. McKinley, Olivier Temam

September 1996 **Proceedings of the seventh international conference on Architectural support for programming languages and operating systems**, Volume 31 , 30 Issue 9 , 5

Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper analyzes and quantifies the locality characteristics of numerical loop nests in order to suggest future directions for architecture and software cache optimizations. Since most programs spend the majority of their time in nests, the vast majority of cache optimization techniques target loop nests. In contrast, the locality characteristics that drive these optimizations are usually collected across the entire application rather than the nest level. Indeed, researchers have studied nume ...

8 Session 17: architecture: Sunder: a programmable hardware prefetch architecture for numerical loops

Tzi-cker Chiueh

November 1994 **Proceedings of the 1994 ACM/IEEE conference on Supercomputing**

Full text available: pdf(922.38 KB) Additional Information: full citation, abstract, references, citings

*Beyond data caching, data prefetching is by far the most effective way to address the memory access bottleneck associated with high-performance processors. This is particularly true for scientific programs whose working sets cannot be easily fit into the on-chip data cache. This paper proposes a new data prefetching architecture called* **Sunder,** *which combines the flexibility and accurateness of software prefetching and the transparency and low-overhead of hardware prefetching. Th ...*

9 Guided region prefetching: a cooperative hardware/software approach
Zhenlin Wang, Doug Burger, Kathryn S. McKinley, Steven K. Reinhardt, Charles C. Weems
May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2
Full text available: pdf(171.47 KB) Additional Information: full citation, abstract, references

Despite large caches, main-memory access latencies still cause significant performance losses in many applications. Numerous hardware and software prefetching schemes have been proposed to tolerate these latencies. Software prefetching typically provides better prefetch accuracy than hardware, but is limited by prefetch instruction overheads and the compiler's limited ability to schedule prefetches sufficiently far in advance to cover level-two cache miss latencies. Hardware prefetching can be e ...

10 A performance study of software and hardware data prefetching schemes
T.-F. Chen, J.-L. Baer
April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture**, Volume 22 Issue 2
Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, citings, index terms

Prefetching, i.e., exploiting the overlap of processor computations with data accesses, is one of several approaches for tolerating memory latencies. Prefetching can be either hardware-based or software-directed or a combination of both. Hardware-based prefetching, requiring some support unit connected to the cache, can dynamically handle prefetches at run-time without compiler intervention. Software-directed approaches rely on compiler technology to insert explicit prefetch instructions. Mowry ...

11 The effectiveness of caches for vector processors
Jeffrey D. Gee, Alan Jay Smith
July 1994 **Proceedings of the 8th international conference on Supercomputing**
Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, index terms

Vector processors have typically used vector registers, interleaved memory, and pipelined access to data to provide sufficient memory system performance. Caches have been used mainly for instructions and scalar data, while vectors are usually uncached, presumably partially because of the belief that there is insufficient vector locality in these workloads. In this study we use memory address traces from an Ardent Titan to examine both reference locality and cache performance in a vector pro ...

12 Vector prefetching
Michael K. Gschwind, Thomas J. Pietsch
December 1995 **ACM SIGARCH Computer Architecture News**, Volume 23 Issue 5
Full text available: pdf(523.08 KB) Additional Information: full citation, abstract, citings, index terms

This paper focuses on extending the memory subsystem by integrating a prefetch buffer mechanism. Prefetching allows high-level application knowledge to increase memory performance, which is currently constraining the performance of most system. While prefetching does not reduce the latency of memory accesses, it hides this latency by overlapping memory access and instruction execution. The first prefetch operation to the buffer is initiated by an explicit fetch instruction. All f ...

13 A scalar architecture for pseudo vector processing based on slide-windowed registers
Hiroshi Nakamura, Taisuke Boku, Hideo Wada, Hiromitsu Imori, Ikuo Nakata, Yasuhiro Inagami, Kisaburo Nakazawa, Yoshiyuki Yamashita
August 1993 **Proceedings of the 7th international conference on Supercomputing**
Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we present a new scalar architecture for high-speed vector processing. Without using cache memory, the proposed architecture tolerates main memory access

latency by introducing slide-windowed floating-point registers with data preloading feature and pipelined memory. The architecture can hold upward compatibility with existing scalar architectures. In the new architecture, software can control the window structure. This is the advantage compared with our previous work of regi ...

14 An effective on-chip preloading scheme to reduce data access penalty
Jean-Loup Baer, Tien-Fu Chen
August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing**
Full text available: pdf(1.18 MB) Additional Information: full citation, references, citings, index terms

15 An effective programmable prefetch engine for on-chip caches
Tien-Fu Chen
December 1995 **Proceedings of the 28th annual international symposium on Microarchitecture**
Full text available: pdf(721.83 KB) Additional Information: full citation, references, citings, index terms

16 Data relocation and prefetching for programs with large data sets
Yoji Yamada, John Gyllenhall, Grant Haab, Wen-mei Hwu
November 1994 **Proceedings of the 27th annual international symposium on Microarchitecture**
Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

Numerical applications frequently contain nested loop structures that process large arrays of data. The execution of these loop structures often produces memory reference patterns that poorly utilize data caches. Limited associativity and cache capacity result in cache conflict misses. Also, non-unit stride access patterns can cause low utilization of cache lines. Data copying has been proposed and investigated in order to reduce cache conflict misses, but this technique has a high executio ...

**Keywords**: cache conflicts, data copying, data relocation, program optimization, software prefetching

17 Reducing cache misses using hardware and software page placement
Timothy Sherwood, Brad Calder, Joel Emer
May 1999 **Proceedings of the 13th international conference on Supercomputing**
Full text available: pdf(1.50 MB) Additional Information: full citation, references, citings, index terms

18 Evaluating stream buffers as a secondary cache replacement
S. Palacharla, R. E. Kessler
April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture**, Volume 22 Issue 2
Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index terms

Today's commodity microprocessors require a low latency memory system to achieve high sustained performance. The conventional high-performance memory system provides fast data access via a large secondary cache. But large secondary caches can be expensive, particularly in large-scale parallel systems with many processors (and thus many caches).We evaluate a memory system design that can be both cost-effective as well as provide better performance, particularly for scientific workloads: a single ...

19 Prefetching using Markov predictors

h c g e cf c

Doug Joseph, Dirk Grunwald
May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2

Full text available: pdf(1.68 MB) Additional Information: full citation, abstract, references, citings, index terms

Prefetching is one approach to reducing the latency of memory operations in modern computer systems. In this paper, we describe the *Markov prefetcher.* This prefetcher acts as an interface between the on-chip and off-chip cache, and can be added to existing computer designs. The Markov prefetcher is distinguished by prefetching *multiple reference predictions* from the memory subsystem, and then prioritizing the delivery of those references to the processor.This design results in a pr ...

**20** A locality sensitive multi-module cache with explicit management
Jesús Sánchez, Antonio González
May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: pdf(2.42 MB) Additional Information: full citation, references, index terms